

## Para saber más: Design Pattern Command

Hemos hablado mucho sobre MVC, que es un patrón de arquitectura. Es decir, un patrón que define una parte de la arquitectura de la aplicación (en nuestro caso, define las capas). Curiosamente, también usamos un Design Pattern para construir el MVC. ¿Cuál?

¡Vamos allá! Tenga en cuenta que cada acción tiene solo un método, que se llama `ejecutar` (`execute`). Veamos el ejemplo:

```
public class NuevaEmpresa implements accion {  
  
    public String ejecutar (HttpServletRequest request, HttpSei  
    }  
}
```

[COPIA EL CÓDIGO](#)

También podríamos haber llamado a este método `run` o `call`, no habría ninguna diferencia, pero siempre el propósito de este método es encapsular la ejecución de la acción.

Nuestra interface `Accion` estandarizó el contrato del método para que todas las acciones sigan el mismo comportamiento.

También tenga en cuenta que, solo por el nombre del método, no sabemos qué se está ejecutando. Para eso, necesitamos mirar el nombre de la clase. Esto tampoco le importa al controlador, lo que importa es que tenga el método `ejecutar`. Siguiendo este patrón, pudimos delegar las llamadas del controlador central hacia las acciones. Genial, pero ¿cuál es el patrón al final

Todas nuestras clases siguen el patrón **Command**, es decir, son comandos siguiendo la nomenclatura del famoso libro de Design Patterns:

- Design Patterns: Elements of Reusable Object-Oriented Software

Los patrones de diseño son un tema importante en el desarrollo de software, ya que permiten resolver un problema de forma estructurada y organizada. Conociendo esta importancia, recomiendo la lectura de ese libro.